Kubernetes:-

Installation can be:-

1. Managed Kubernetes services

2. Minikube (Single-Node Kubernetes Cluster)

3. Installing Kubernetes & Configure Manually(Hard way)

Kubectl commands:-

For launching the container in the pod=>

#kubectl run mywebserver –image=nginx

For verifying the configured container inside the pod=>

#kubectl get pods

For running the container in interactive mode=>

#kubectl exec -it mywebserver – bash

For running any command inside the container=>

#kubectl exec -it mywebserver –ls -l /

For removing the pod(Pod will be deleted even if it is running)=>

#kuberetes delete pod mywebserver

For building the application by the yaml file=>

#kubectl apply -f pod.yaml

Delete Everything inside the file=>

#kubectl delete -f pod.yaml

Getting the details of the pod=>

#kubectl desc pod “pod\_name”

Expose the port of pod=>

#kubectl run ngnix –image=nginx –port=80

Output the manifest file=>

#kubectl run nginx –image=nginx –port=80 –dry-run=client(actually not create the object so used for testing purpose) -o yaml(get the yaml file)

Deleting all the pod that are in the cluster=>

#kubectl delete pod --all

Get all the labels attached to cluster=>

#kubectl get pods –show-labels

Check for the replicaset=>

#kubectl get replicaset

Check for Deployment=>

#kubectl get deployment

Check for all the available version for rollingout=>

#kubectl rollout history deployment.v1.apps/kplabs\_deployment

For rolling back to the specific revision from the history=>

#kubectl rollout undo deployment.v1.apps/kplabs\_deployment --to-revision=2

For creating the deployment by cli=>

#kubectl create deployment my-dep –image=ngnix –replicas 3

We can delete the deployment=>

#kubectl delete deployment my\_deployment

For any help related to deployment=>

#kubectl create deployment –help

Set the image to the deployment=>

#kubectl set image deployment ngnix nginx=nginx:latest

--record instruction for storing the image=>

#kubectl set image deployment nginx nginx=nginx:1.9.1 –record

Rolling back to the previous revision=>

#kubectl rollout undo deployment nginx

Scaling in the deployment=>

#kubectl scale deployment nginx –replicas 5

Get the deployment manifest file by cli=>

#kubectl create deployment nginx-deploy –image=nginx --dry-run=client -o yaml

Check the pods are launched on which node/IP

#kubectl get pods -o wide

Get the daemonset =>

#kubectl get daemonset (the desired pods are equivalent to the no. of worker node)

Getting the details about the daemonset=>

#kubectl describe daemonset kplabs\_daemonset

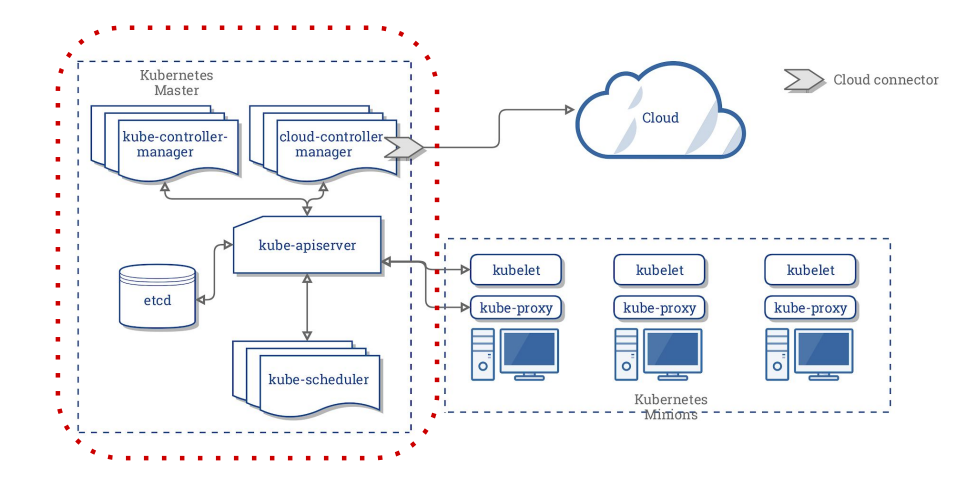
POD:- Represents the group of application container that may shared the same resources.

Pods are useful in case of multiple dependent containers in the specific application.

Pod has the specific port and IP Address but the containers inside it shares the same IP & Port.

Kubernetes Objects:- It refers to the record of intend, If the object is created, Kubernetes ensures for its working and existence of that object.

Kubernetes Architecture:-



Kubernetes Master Components:-

1. Kube-apiserver

2. etcd

3. kube-scheduler

4. kube-controller-manager

5. cloud-controller-manager

Kubernetes Worker Components:-

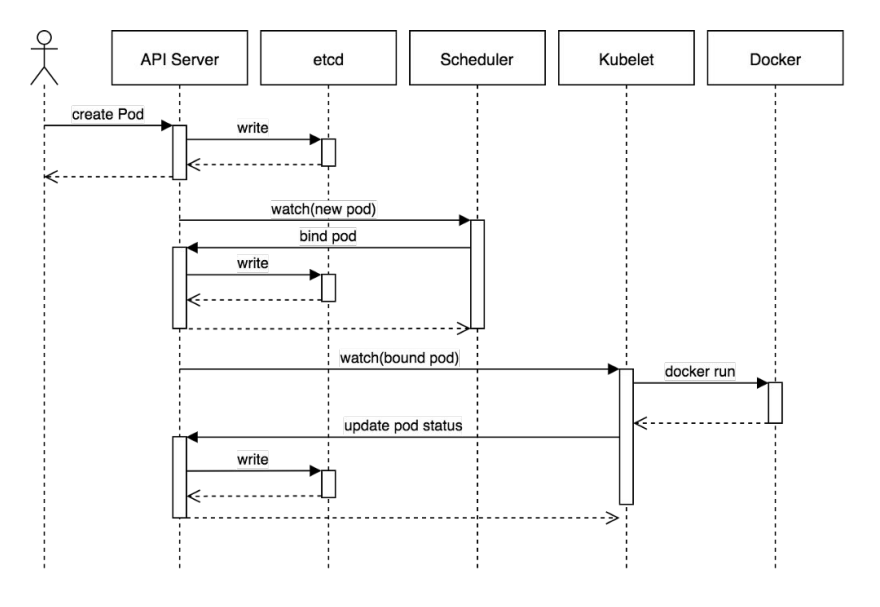
1. Kubelets

2. kube-proxy

ETCD:- Distributed Reliable Key-value storage. It store all the configuration data of the Kubernetes cluster like o. of pod available, no. of containers running, etc.

Kube-Apiserver:- It acts as a gateway to Kubernetes Cluster. Kubectl is required for any command by user and kubectl in returns connects to Apiserver so, For any kind of interaction in the Kubernetes cluster Apiserver is main.

Flow Diagram of Kubernetes:-



Kube-Scheduler:-It will assign the node to the pod according to requirements.

Exposing the port in the Kubernetes can be done by containerPort field at port parameter of the container in yaml file.

Labels & Selectors:-

Labels are the Key-value pairs attach to objects(pods), Adding label to the resource make it easy to understand.

Selectors are used as the filter based on the assigned labels. Selectors make it easier to access the desired pod & do the operations.

ReplicaSet:-

Maintain the current running no. of pods to the desired running no. of pods.

Deployment:-

Unlike Replicaset, it also facilitate rollingout & rolling back the changes done by the replicas.

Hirarchy:- Pods managed by replicaset and replicasets by deployment.

DEPLOYMENT

REPLICASETS

PODS

MaxSurge 25% strategy:- It refers to the maximum no. of pods that can become unavailable during the deployment is 25%.

--record instruction is used for storing the revision of deployment.

DaemonSet:-

DaemonSet ensures that all nodes runs the copy of pod. As the node is added to the cluster the pods are also added to it.

NodeSelector:-

It allows us to add the constraint to run the pod on the specific worker node. We can write the parameter nodeSelector at yaml file for putting this constraint.